
Simultaneous Object Detection, Tracking, and Event Recognition

Andrei Barbu, Aaron Michaux, Siddharth Narayanaswamy, and Jeffrey Mark Siskind

Purdue University
 School of Electrical and Computer Engineering
 465 Northwestern Avenue
 West Lafayette IN 47907-2035 USA
 andrei@0xab.com, {amichaux, snarayan, qobi}@purdue.edu

Abstract

The common internal structure and algorithmic organization of object detection, detection-based tracking, and event recognition facilitates a general approach to integrating these three components. This supports multidirectional information flow between these components allowing object detection to influence tracking and event recognition and event recognition to influence tracking and object detection. The performance of the combination can exceed the performance of the components in isolation. This can be done with linear asymptotic complexity.

1 Introduction

Many common approaches to event recognition (Siskind and Morris, 1996; Starner et al., 1998; Wang et al., 2009; Xu et al., 2002, 2005) classify events based on their motion profile. This requires detecting and tracking the event participants. Adaptive approaches to tracking (Yilmaz et al., 2006), e.g. Kalman filtering (Comaniciu et al., 2003), suffer from three difficulties that impact their utility for event recognition. First, they must be initialized. One cannot initialize on the basis of motion since many event participants move only for a portion of the event, and sometimes not at all. Second, they exhibit drift and often must be

periodically reinitialized to compensate. Third, they have difficulty tracking small, deformable, or partially occluded objects as well as ones whose appearance changes dramatically. This is particularly of concern since many events, e.g. picking things up, involve humans interacting with objects that are sufficiently small for humans to grasp and where such interaction causes appearance change by out-of-plane rotation, occlusion, or deformation.

Detection-based tracking is an alternate approach that attempts to address these issues. In detection-based tracking an object detector is applied to each frame of a video to yield a set of candidate detections which are composed into tracks by selecting a single candidate detection from each frame that maximizes temporal coherency of the track. However, current object detectors are far from perfect. On the PASCAL VOC Challenge, they typically achieve average precision scores of 40% to 50% (Everingham et al., 2010). Directly applying such detectors on a per-frame basis would be ill-suited to event recognition. Since the failure modes include both false positives and false negatives, interpolation does not suffice to address this shortcoming. A better approach is to combine object detection and tracking with a single objective function that maximizes temporal coherency to allow object detection to inform the tracker and vice versa.

One can carry this approach even further and integrate event recognition with both object detection and tracking. One way to do this is to incorporate coherence with a target event model into the temporal coherency measure. For example, a top-down expectation of observing a *pick up* event can bias the object detector and tracker to search for event

Additional images and videos as well as all code and datasets are available at <http://engineering.purdue.edu/~qobi/arxiv2012a>.

participants that exhibit the particular joint motion profile of that event: an object in close proximity to the agent, the object starting out at rest while the agent approaches the object, then the agent touching the object, followed by the object moving with the agent. Such information can also flow bidirectionally. Mutual detection of a *baseball bat* and a *hitting* event can be easier than detecting each in isolation or having a fixed direction of information flow.

The common internal structure and algorithmic organization of current object detectors (Felzenszwalb et al., 2010a,b), detection-based trackers (Wolf et al., 1989), and HMM-based approaches to event recognition (Baum and Petrie, 1966) facilitates a general approach to integrating these three components. We demonstrate an approach to integrating object detection, tracking, and event recognition and show how it improves each of these three components in isolation. Further, while prior detection-based trackers exhibit quadratic complexity, we show how such integration can be fast, with linear asymptotic complexity.

2 Detection-based tracking

The methods described in sections 4, 5, and 6 extend a popular dynamic-programming approach to detection-based tracking. We review that approach here to set forth the concepts, terminology, and notation that will be needed to describe the extensions.

Detection-based tracking is a general framework where an object detector is applied to each frame of a video to yield a set of candidate detections which are composed into tracks by selecting a single candidate detection from each frame that maximizes temporal coherency of the track. This general framework can be instantiated with answers to the following questions:

1. What is the representation of a *detection*?
2. What is the *detection source*?
3. What is the measure of temporal coherency?
4. What is the procedure for finding the track with maximal temporal coherency?

We answer questions 1 and 2 by taking a detection to be a scored axis-aligned rectangle (box), such as produced by the Felzenszwalb et al. (2010a,b) object detectors, though our approach is compatible with any method for producing scored axis-aligned rectangular detections. If b_j^t denotes the j th detection in frame t , $f(b_j^t)$ denotes the score of that detection, T denotes the number of frames, and $\mathbf{j} = \langle j_1, \dots, j_T \rangle$ denotes a track comprising the j_t th detection in frame t , we answer question 3 by formulating temporal coherency of a track $\mathbf{j} = \langle j_1, \dots, j_T \rangle$ as:

$$\max_{j_1, \dots, j_T} \sum_{t=1}^T f(b_{j_t}^t) + \sum_{t=2}^T g(b_{j_{t-1}}^{t-1}, b_{j_t}^t) \quad (1)$$

where g scores the local temporal coherency between de-

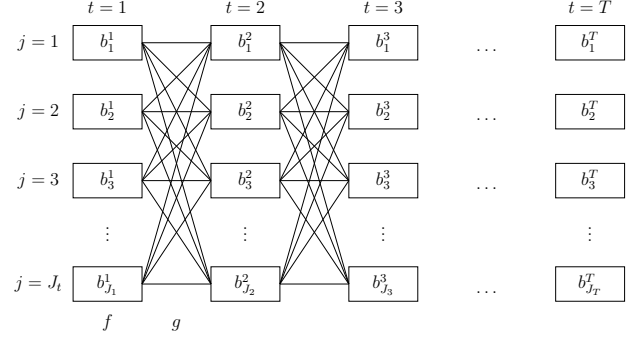


Figure 1: The tracking lattice constructed by the Viterbi algorithm performing detection-based tracking.

tections in adjacent frames. We take g to be the negative Euclidean distance between the center of $b_{j_t}^t$ and the center of $b_{j_{t-1}}^{t-1}$ projected forward one frame, though, as discussed below, our approach is compatible with a variety of functions discussed by Felzenszwalb and Huttenlocher (2004). The forward projection internal to g can be done in a variety of ways including optical flow and the Kanade-Lucas-Tomasi (KLT) (Shi and Tomasi, 1994; Tomasi and Kanade, 1991) feature tracker. We answer question 4 by observing that Eq. 1 can be optimized in polynomial time with the Viterbi algorithm (Viterbi, 1971):

$$\begin{aligned} &\text{for } j = 1 \text{ to } J_1 \text{ do } \delta_j^1 := f(b_j^1) \\ &\text{for } t = 2 \text{ to } T \\ &\quad \text{do for } j = 1 \text{ to } J_t \\ &\quad \quad \text{do } \delta_j^t := f(b_j^t) + \max_{j'=1}^{J_{t-1}} g(b_{j'}^{t-1}, b_j^t) + \delta_{j'}^{t-1} \end{aligned} \quad (2)$$

where J_t is the number of detections in frame t . This leads to a lattice as shown in Fig. 1.

Detection-based trackers exhibit less drift than adaptive approaches to tracking due to fixed target models. They also tend to perform better than simply picking the best detection in each frame. The reason is that one can allow the detection source to produce multiple candidates and use the combination of the detection score f and the adjacent-frame temporal-coherency score g to select the track. The essential attribute of detection-based tracking is that g can overpower f to assemble a more coherent track out of weaker detections. The nonlocal nature of Eq. 1 can allow more-reliable tracking with less-reliable detection sources.

A crucial practical issue arises: *How many candidate detections should be produced in each frame?* Producing too few may risk failing to produce the desired detection that is necessary to yield a coherent track. In the limit, it is impossible to construct any track if even a single frame lacks any detections. The current state-of-the-art in object detection is unable to simultaneously achieve high precision

One can ameliorate this somewhat by constructing a lattice that skips frames (Sala et al., 2010). This increases the asymp-

and recall and thus it is necessary to explore the trade-off between the two (Everingham et al., 2010). A detection-based tracker can bias the detection source to yield higher recall at the expense of lower precision and rely on temporal coherency to compensate for the resulting lower precision. This can be done in at least three ways. First, one can depress the detection-source acceptance thresholds. One way this can be done with the Felzenszwalb et al. detectors is to lower the trained model thresholds. Second, one can pool the detections output by multiple detection sources with complementary failure modes. One way this can be done is by training multiple models for people in different poses. Third, one can use adaptive-tracking methods to project detections forward to augment the raw detector output and compensate for detection failure in subsequent frames. This can be done in a variety of ways including optical flow and KLT. The essence of our paper is a more principled collection of approaches for compensating for low recall in the object detector.

A practical issue arises when pooling the detections output by multiple detection sources. It is necessary to normalize the detection scores for such pooled detections by a per-model offset. One can derive an offset by computing a histogram of scores of the top detection in each frame of a video and taking the offset to be the minimum of the value that maximizes the between-class variance (Otsu, 1979) when bipartitioning this histogram and the trained acceptance threshold offset by a small but fixed amount.

The operation of a detection-based tracker is illustrated in Fig. 2. This example demonstrates several things of note. First, reliable tracks are produced despite an unreliable detection source. Second, the optimal track contains detections with suboptimal score. Row (b) demonstrates that selecting the top-scoring detection does not yield a temporally-coherent track. Third, forward-projection of detections from the second to third column in row (c) compensates for the lack of raw detections in the third column of row (a).

Detection-based tracking runs in time $O(TJ^2)$ on videos of length T with J detections per frame. In practice, the run time is dominated by the detection process and the dynamic-programming step. Limiting J to a small number speeds up the tracker considerably while minimally impacting track quality. We further improve the speed of the detectors when running many object classes by factoring the computation of the HOG pyramid.

3 Evaluation of detection-based tracking

We evaluated detection-base tracking using the year-one (Y1) corpus produced by DARPA for the Mind’s Eye pro-

totic complexity to be exponential in the number of frame skips allowed.

gram. These videos are provided at 720p@30fps and range from 42 to 1727 frames in length, with an average of 438.84 frames, and depict people interacting with a variety of objects to enact common English verbs.

Four Mind’s Eye teams (University at Buffalo, Corso 2011, Stanford Research Institute, Bui 2011, University of California at Berkeley, Saenko 2011, and University of Southern California, Navatia 2011) independently produced human-annotated tracks for different portions of Y1. We used these sources of human-annotated tracks to evaluate the performance of detection-based tracking by computing human-human intercoder agreement between all pairs of the four sources of human-annotated tracks and human-machine intercoder agreement between a detection-based tracker and all four of these sources. Since each team annotated different portions of Y1, each such intercoder agreement measure was computed only over the N videos shared by each pair, as reported in Table 1(a). One team (University at Buffalo, Corso 2011) annotated detections as clusters of quadrilaterals around object parts. These were converted to a single bounding box.

Different teams labeled the tracks with different class labels. It was possible to determine from these labels whether the track was for a person or nonperson by assuming that the labels ‘person’ and ‘human’, and only those labels, denoted person tracks, but it was not possible to automatically make finer-grained class comparisons. Thus we independently compared person tracks with person tracks and nonperson tracks with nonperson tracks. When comparing an annotation u of a video n containing l_n^u person tracks with an annotation v of that same video containing l_n^v person tracks, we compared $[l_n^u, l_n^v]$ person tracks. We selected the best over all $[l_n^u, l_n^v]!$ permutation mappings ρ_n between person tracks in u and person tracks in v . A permutation mapping was preferred when it had higher average overlap score among corresponding boxes across the tracks and the frames in a video, where the overlap score was that used by the PASCAL VOC Challenge (Everingham et al., 2010), namely the ratio of the area of their intersection to the area of their union. Different tracks could annotate different frames of a video. When comparing such, we considered only the shared frames.

For every pair of teams, we computed the mean and standard deviation of the overlap score across all shared frames in all tracks in the best permutation mappings for all shared videos. The averaging process used both to determine the best permutation mapping for each video pair and to determine overall mean and standard deviation measures weighted each overlap score equally. More precisely, if $1 \leq n \leq N$, $1 \leq l \leq [l_n^u, l_n^v]$ denotes a shared track for video n , T_n^l denotes the set of shared frames for that shared track l in video n , U_t^n and V_t^n denote the vector of boxes for frame t in video n for annotations u and v respectively, and O denotes the overlap measure, we score a permutation

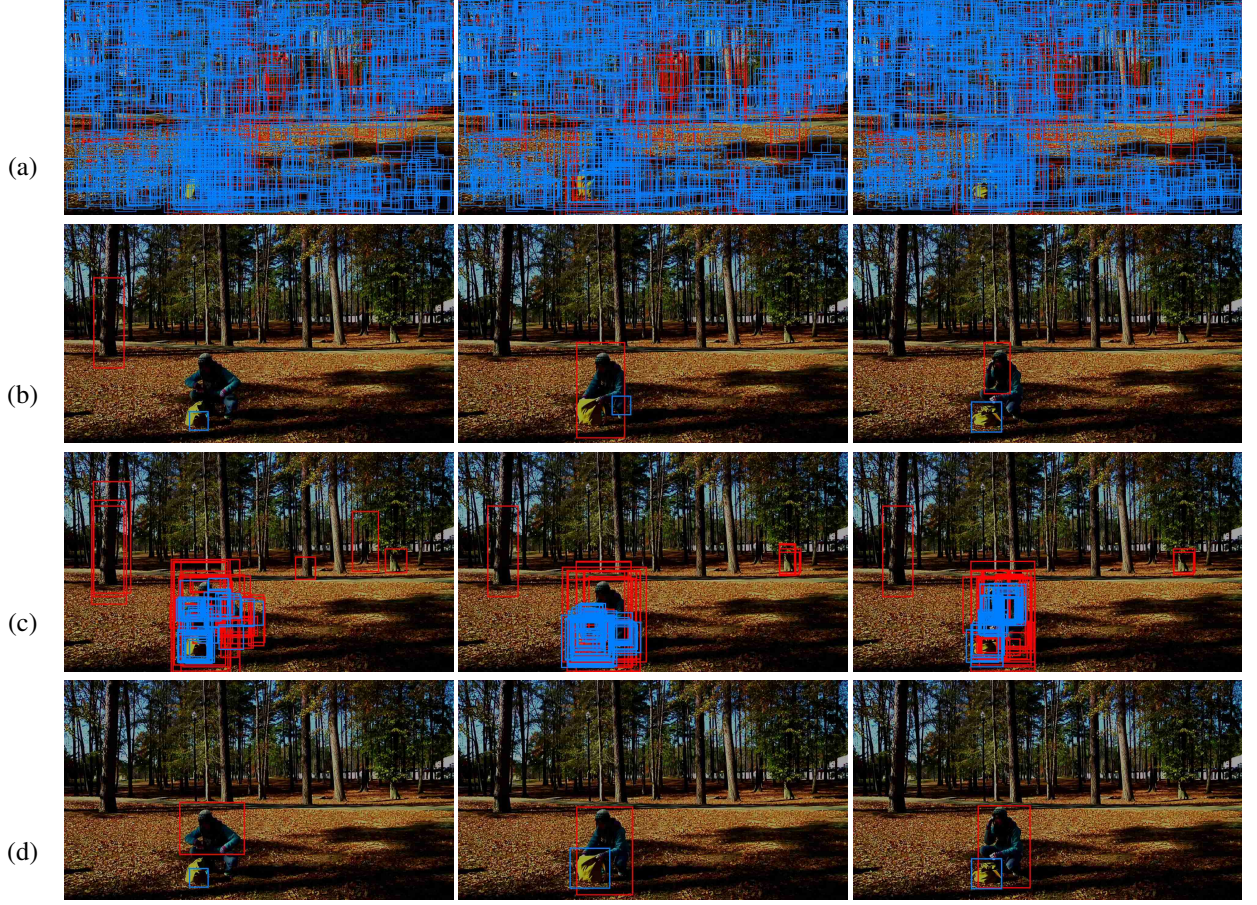


Figure 2: The operation of a detection-based tracker. (a) Output of the detection sources, biased to yield false positives. (b) The top-scoring output of the detection source. (c) Augmenting the output of the detection sources with forward-projected detections. (d) The optimal tracks selected by the Viterbi algorithm.

mapping ρ_n for video n as:

$$\frac{1}{\sum_{l=1}^{|T_n^u|} |T_n^l|} \sum_{l=1}^{|T_n^u|} \sum_{t \in T_n^l} O(\rho_n(U_t^n)[l], V_t^n[l])$$

and computed the mean overlap for a pair of teams as:

$$\frac{1}{\sum_{n=1}^N \sum_{l=1}^{|T_n^u|} |T_n^l|} \sum_{n=1}^N \sum_{l=1}^{|T_n^u|} \sum_{t \in T_n^l} O(\rho_n(U_t^n)[l], V_t^n[l])$$

with an analogous computation for standard deviation and nonperson tracks.

The overall mean and standard deviation measures, reported in Table 1(b,c), indicate that the mean human-human overlap is only marginally greater than the mean human-machine overlap by about one standard deviation. This suggests that improvement in tracker performance is

unlikely to lead to significant improvement in action recognition performance and sentential description quality.

4 Combining object detection and tracking

While detection-based tracking is resilient to low precision, it requires perfect recall; it cannot generate a track through a frame that has no detections and it cannot generate a track through a portion of the field of view which has no detections regardless of how good the temporal-coherence of the resulting track would be. This brittleness means that any detection source employed will have to significantly over-generate detections to achieve near-perfect recall. This has a downside. While the Viterbi algorithm has linear complexity in the number of frames, it is quadratic in the number of detections per frame. This drastically limits the number of detections that can reasonably be processed leading to the necessity of tuning the thresholds on the detection sources. We have developed a novel mechanism to eliminate the need for a threshold and track every possible detection, at every position and scale in the image, in

N	UB	SRI	UCB	USC	μ	UB	SRI	UCB	USC	σ	UB	SRI	UCB	USC
UB		8	20	8	UB		0.76	0.68	0.59	UB		0.06	0.14	0.10
SRI	8		1201	95	SRI	0.76		0.55	0.59	SRI	0.06		0.27	0.16
UCB	20	1201		204	UCB	0.68	0.55		0.48	UCB	0.14	0.27		0.23
USC	8	95	204		USC	0.59	0.59	0.48		USC	0.10	0.16	0.23	
us	48	1254	1829	360	us	0.54	0.40	0.35	0.43	us	0.26	0.24	0.23	0.20
(a)					(b)					(c)				

Table 1: (a) The number of videos in common, (b) the mean overlap, and (c) the standard deviation in overlap between each pair of annotation sources.

time linear in the number of detections and frames. At the same time our approach eliminates the need for forward projection since every detection is already present. Our approach involves simultaneously performing object detection and tracking, optimizing the joint object-detection and temporal-coherency score.

Our general approach is to compute the distance between pairs of detection pyramids for adjacent frames, rather than using g to compute the distance between pairs of individual detections. These pyramids represent the set of all possible detections at all locations and scales in the associated frame. Employing a distance transform makes this process linear in the number of location and scale positions in the pyramid. Many detectors, e.g. those of Felzenszwalb et al., use such a scale-space representation of frames to represent detections internally even though they might not output such. Our approach requires instrumenting such a detector to provide access to this internal representation.

At a high-level, the Felzenszwalb et al. detectors learn a forest of HOG (Freeman and Roth, 1995) filters for each object class along with their characteristic displacements. Detection proceeds by applying each HOG filter at every position in an image pyramid followed by computing the optimal displacements at every position in that image pyramid, thereby creating a new pyramid, the detection pyramid. Finally, the detector searches the detection pyramid for high-scoring detections and extracts those above a threshold. The detector employs a dynamic-programming algorithm to efficiently compute the optimal part displacements for the entire image pyramid. This algorithm (Felzenszwalb et al., 2010a) is very similar to the Viterbi algorithm. It is made tractable by the use of a generalized distance transform (Felzenszwalb and Huttenlocher, 2004) that allows it to scale linearly with the number of image pyramid positions. Given a set \mathcal{G} of points (which in our case denotes an image pyramid), a distance metric d between pairs of points p and q , and an arbitrary function $\phi : \mathcal{G} \rightarrow \mathbb{R}$, the generalized distance transform $D_\phi(q)$ computes:

$$D_\phi(q) = \min_{p \in \mathcal{G}} (d(p, q) + \phi(q))$$

in linear time for certain distance metrics including squared Euclidean distance.

Instead of extracting and tracking just the thresholded detections, one can directly track all detections in the entire pyramid simultaneously by defining a distance measure between detection pyramids for adjacent frames and performing the Viterbi tracking algorithm on these pyramids instead of sets of detections in each frame. To allow comparison between detections at different scales in the detection pyramid, we convert the detection pyramid to a rectangular prism by scaling the coordinates of the detections at scale s by $\pi(s)$, chosen to map the detection coordinates back to the coordinate system of the input frame. We define the distance between two detections, b and b' , in two detection pyramids as a scaled squared Euclidean distance:

$$d(b_{xys}, b'_{x'y's'}) = (\pi(s)x - \pi(s')x')^2 + (\pi(s)y - \pi(s')y')^2 + \alpha(s - s')^2 \quad (3)$$

where x and y denote the original image coordinates of a detection center at scale s . Nominally, detections are boxes. Comparing two such boxes involves a four-dimensional distance metric. However, with a detection pyramid, the aspect ratio of detections is fixed, reducing this to a three-dimensional distance metric. The coefficient α in the distance metric weights a difference in detection area differently than detection position.

The above amounts to replacing detections b_j^t with b_{xys}^t , lattice values δ_j^t with δ_{xys}^t , and Eq. 2 with:

$$\begin{aligned} &\text{for } x = 1 \text{ to } X \\ &\quad \text{do for } y = 1 \text{ to } Y \\ &\quad\quad \text{do for } s = 1 \text{ to } S \text{ do } \delta_{xys}^1 := f(b_{xys}^1) \\ &\quad \text{for } t = 2 \text{ to } T \\ &\quad\quad \text{do for } x = 1 \text{ to } X \\ &\quad\quad\quad \text{do for } y = 1 \text{ to } Y \\ &\quad\quad\quad\quad \text{do for } s = 1 \text{ to } S \\ &\quad\quad\quad\quad\quad \text{do } \delta_{xys}^t := f(b_{xys}^t) \\ &\quad\quad\quad\quad\quad\quad + \max_{x', y', s'} (g(b_{x'y's'}^{t-1}, b_{xys}^t) + \delta_{x'y's'}^{t-1}) \end{aligned} \quad (4)$$

The above formulation allows us to employ the generalized distance transform as an analog to g in Eq. 1, although it restricts consideration of g to be squared Euclidean distance rather than Euclidean distance. We avail ourselves of the

fact that the generalized distance transform operates independently on each of the three dimensions x , y , and s in order to incorporate α into Eq. 3. While linear-time use of the distance transform restricts the form of g , it places no restrictions on the form of f .

One way to view the above is that the vector of δ_j^t for all $1 \leq j \leq J_t$ from Eq. 2 is being represented as a pyramid and the loop:

$$\begin{aligned} &\text{for } j = 1 \text{ to } J_t \\ &\text{do } \delta_j^t := f(b_j^t) + \max_{j'=1}^{J_{t-1}} g(b_{j'}^{t-1}, b_j^t) + \delta_{j'}^{t-1} \end{aligned} \quad (5)$$

is being performed as a linear-time construction of a generalized distance transform rather than a quadratic-time nested pair of loops. Another way to view the above is that we generalize the notion of a detection pyramid from representing per-frame detections b_{xys} at three-dimensional pyramid positions $\langle x, y, s \rangle$ to representing per-video detections b_{xys}^t at four-dimensional pyramid positions $\langle x, y, s, t \rangle$ and finding a sequence of per-video detections for $1 \leq t \leq T$ that optimizes the following variant of Eq. 1:

$$\max_{\substack{x_1, \dots, x_T \\ y_1, \dots, y_T \\ s_1, \dots, s_T}} \sum_{t=1}^T f(b_{x_t y_t s_t}^t) + \sum_{t=2}^T g(b_{x_{t-1} y_{t-1} s_{t-1}}^{t-1}, b_{x_t y_t s_t}^t) \quad (6)$$

This combination of the detector and the tracker is performing simultaneous detection and tracking integrating the information between the two. Before, the tracker was affected by the detector but the detector was unaffected by the tracker: potential low-scoring but temporally-coherent detections would not even be generated by the detector despite the fact that they would yield good tracks. Because now, the detector no longer chooses which detections to produce but instead scores all detections at every position and scale, the tracker is able to choose among any possible detection. Such tight integration of higher- and lower-level information will be revisited when integrating event models into this framework.

5 Combining tracking and event detection

It is popular to use Hidden Markov Models (HMMs) to perform event recognition (Siskind and Morris, 1996; Starner et al., 1998; Wang et al., 2009; Xu et al., 2002, 2005). When doing so, the log likelihood of a video conditioned on an event model is:

$$\log \sum_{k_1, \dots, k_T} \exp \sum_{t=1}^T h(k_t, b_{j_t}^t) + \sum_{t=2}^T a(k_t, k_{t-1})$$

where k_t denotes the state of the HMM for frame t , $h(k, b)$ denotes the log probability of generating a detection b conditioned on being in state k , $a(k, k')$ denotes the log probability of transitioning from state k to k' , and j_t^* denotes

index of the detection produced by the tracker in frame t . This log likelihood can be computed with the forward algorithm (Baum and Petrie, 1966) which is analogous to the Viterbi algorithm. Maximum likelihood (ML), the standard approach to using HMMs for classification, selects the event model that maximizes the likelihood of an observed event. One can instead select the model with the maximum *a posteriori* (log) probability (MAP).

$$\max_{k_1, \dots, k_T} \sum_{t=1}^T h(k_t, b_{j_t^*}^t) + \sum_{t=2}^T a(k_t, k_{t-1}) \quad (7)$$

This can be computed with the Viterbi algorithm. The advantage of doing so is that one can combine the Viterbi algorithm used for detection-based tracking with the Viterbi algorithm used for event classification.

One can combine Eq. 1 with Eq. 7 to yield a unified cost function:

$$\begin{aligned} \max_{j_1, \dots, j_T} \max_{k_1, \dots, k_T} & \sum_{t=1}^T f(b_{j_t}^t) + \sum_{t=2}^T g(b_{j_{t-1}}^{t-1}, b_{j_t}^t) \\ & + \sum_{t=1}^T h(k_t, b_{j_t}^t) + \sum_{t=2}^T a(k_t, k_{t-1}) \end{aligned} \quad (8)$$

that computes the joint MAP of the best possible track and the best possible state sequence by replacing j_t^* with j_t inside nested quantification. This too can be computed with the Viterbi algorithm, taking the lattice values δ_{jk}^t to be indexed by the detection index j and the state k , forming the cross product of the tracker lattice nodes and the event lattice nodes:

$$\begin{aligned} &\text{for } j = 1 \text{ to } J_1 \\ &\text{do for } k = 1 \text{ to } K \text{ do } \delta_{jk}^1 := f(b_j^1) + h(k, b_j^1) \\ &\text{for } t = 2 \text{ to } T \\ &\text{do for } j = 1 \text{ to } J_t \\ &\text{do for } k = 1 \text{ to } K \\ &\text{do } \delta_{jk}^t := f(b_j^t) + h(k, b_j^t) \\ &\quad + \max_{j'=1}^{J_{t-1}} \max_{k'=1}^K g(b_{j'}^{t-1}, b_j^t) + a(k, k') \\ &\quad + \delta_{j'k'}^{t-1} \end{aligned} \quad (9)$$

This finds the optimal path through a graph where the nodes at every frame represent the cross product of the detections and the HMM states.

Doing so performs simultaneous tracking and event classification. Before, the event classifier was affected by the tracker but the tracker was unaffected by the event classifier: potential low-scoring tracks would not even be generated by the tracker despite the fact that they would yield a high MAP estimate for some event class. Because now, the tracker no longer chooses which tracks to produce but instead scores all tracks, the event classifier is able to choose among any possible track. This amounts to a different kind of track-coherence measure that is tuned to specific events.

Such a measure might otherwise be difficult to achieve without top-down information from the event classifier. For example applying this method to a video of a running person along with an event model for running, will be more likely to compose a track out of person detections that has high velocity and low change in direction.

Processing each frame t with the algorithm in Eq. 9 is quadratic in $J_t K$. This can be problematic since $J_t K$ can be large. As before, we can make this linear in J_t using a generalized distance transform. One can make this linear in K for suitable state-transition functions a (Felzenszwalb et al., 2003).

Two practical issues arise when applying the above method. First, one can factor Eq. 10 as Eq. 11:

$$\max_{j'=1}^{J_t-1} \max_{k'=1}^K \left(g(b_{j'}^{t-1}, b_j^t) + a(k, k') + \delta_{j'k'}^{t-1} \right) \quad (10)$$

$$\max_{j'=1}^{J_t-1} \left(g(b_{j'}^{t-1}, b_j^t) + \max_{k'=1}^K \left(a(k, k') + \delta_{j'k'}^{t-1} \right) \right) \quad (11)$$

This is important because the computation of $g(b_{j'}^{t-1}, b_j^t)$ might be expensive as it involves a projection of $b_{j'}^{t-1}$ forward one frame (e.g. using optical flow or KLT). Second, when applying this method to multiple event models, the same factorization can be extended to cache the computation of $g(b_{j'}^{t-1}, b_j^t)$ across different event models as this term does not depend on the event model.

6 Combining object detection, tracking and event detection

One can combine the methods of Sections 4 and 5 to optimize a cost function:

$$\max_{\substack{x_1, \dots, x_T \\ y_1, \dots, y_T \\ s_1, \dots, s_T \\ k_1, \dots, k_T}} \sum_{t=1}^T f(b_{x_t y_t s_t}^t) + h(k_t, b_{x_t y_t s_t}^t) + \sum_{t=2}^T g(b_{x_{t-1} y_{t-1} s_{t-1}}^{t-1}, b_{x_t y_t s_t}^t) + a(k_t, k_{t-1}) \quad (12)$$

that combines Eq. 6 with Eq. 8 by forming a large Viterbi lattice with values δ_{xysk}^t .

One practical issue arises when applying the above method. In Eq. 12, h is a function of $b_{x_t y_t s_t}^t$, the detection in the current frame. This allows the HMM event model to depend on static object characteristics such as position, shape, and pose. However, many approaches to event recognition using HMMs use temporal derivatives of such characteristics to provide object velocity and acceleration information (Siskind and Morris, 1996; Starner et al., 1998). Having h also be a function of $b_{x_{t-1} y_{t-1} s_{t-1}}^{t-1}$, the detection in the previous frame, requires incorporation h into the generalized distance transform and thus restricts its form.

The above combination performs simultaneous object detection, tracking, and event classification, integrating information across all three. Without such information integration, the object detector is unaffected by the tracker which is in turn unaffected by the event model. With such integration, the event model can influence the tracker and both can influence the object detector.

This is important because current object detectors cannot reliably detect small, deformable, or partially occluded objects. Moreover, current trackers also fail to track such objects. Information from the event model can focus the object detector and tracker on those particular objects that participate in a specified event. An event model for recognizing an agent picking an object up can bias the object detector and tracker to search for an object that exhibits a particular profile of motion relative to the agent, namely where the object is in close proximity to the agent, the object starts out being at rest while the agent approaches the object, then the agent touches the object, followed by the object moving with the agent.

A traditional view of the relationship between object and event detection suggests that one recognizes a *hammering* event, in part, because one detects a *hammer*. Our unified approach inverts the traditional view, suggesting that one can recognize a *hammer*, in part, by detecting a *hammering* event. Furthermore, a strength of our approach is that such relationships are not encoded explicitly, do not have to be annotated in the training data for the event models, and are learned automatically as part of learning the parameters of the different event models. This is to say that the relationship between a person and the objects they manipulate can be learned from the co-occurrence of tracks in the training data, rather than from manually annotated symbolic relationships.

7 Experimental results

Figure 3 demonstrates improved performance of simultaneous object detection and tracking (c) over object detection (a) and tracking (b) in isolation. This happens for different reasons: motion blur, even for large objects, can lead to poor detection results and hence poor tracks, small objects are difficult to detect and track, and integration can improve detection and tracking of deformable objects, such as a person transitioning from an upright pose to sitting down.

Figure 4 demonstrates improved performance of simultaneous tracking and event recognition (c) over tracking (b) in isolation. These results were obtained with object and event models that were trained independently. The object

It would appear possible to co-train object and event models by combining Baum-Welch (Baum, 1972; Baum et al., 1970) with the training procedure for object models (Felzenszwalb et al., 2010a).

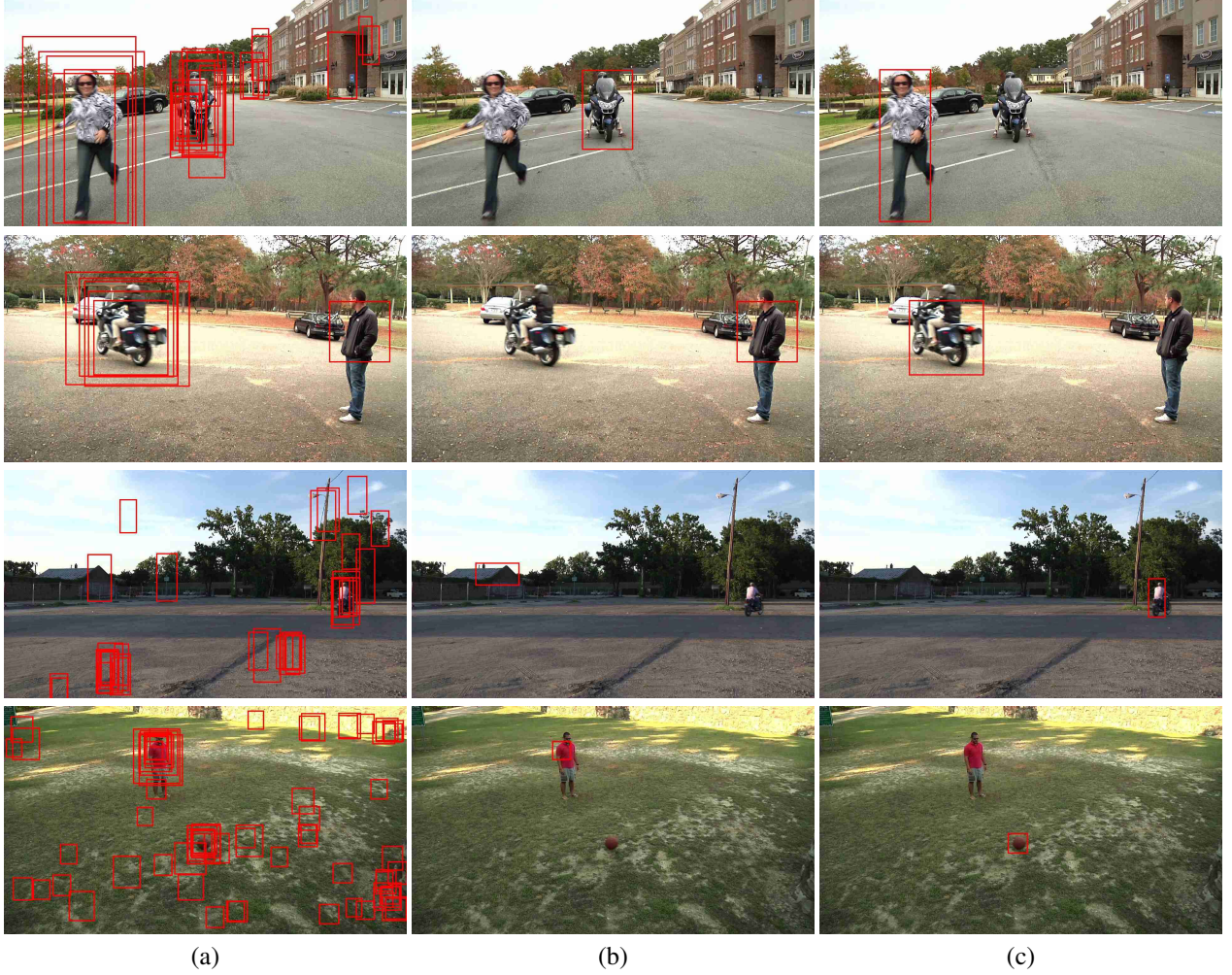


Figure 3: Improved performance of simultaneous object detection and tracking. (a) Output of the Felzenszwalb et al. detector. (b) Tracks produced by detection-based tracking. (c) Tracks produced by simultaneous object detection and tracking.

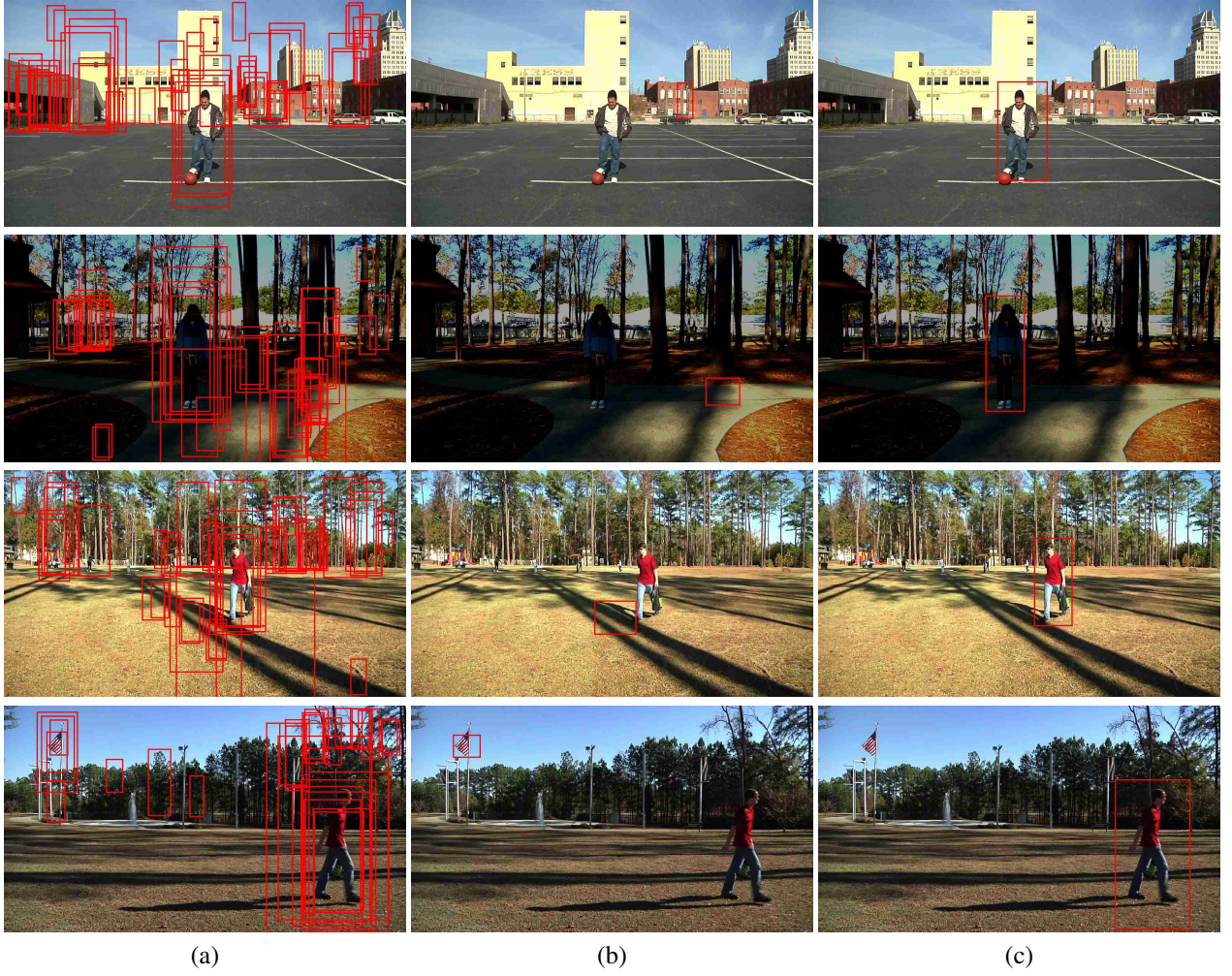


Figure 4: Improved performance of simultaneous tracking and event recognition. (a) Output of the Felzenszwalb et al. detector. (b) Tracks produced by detection-based-tracking. (c) Tracks produced by simultaneous tracking and event recognition.

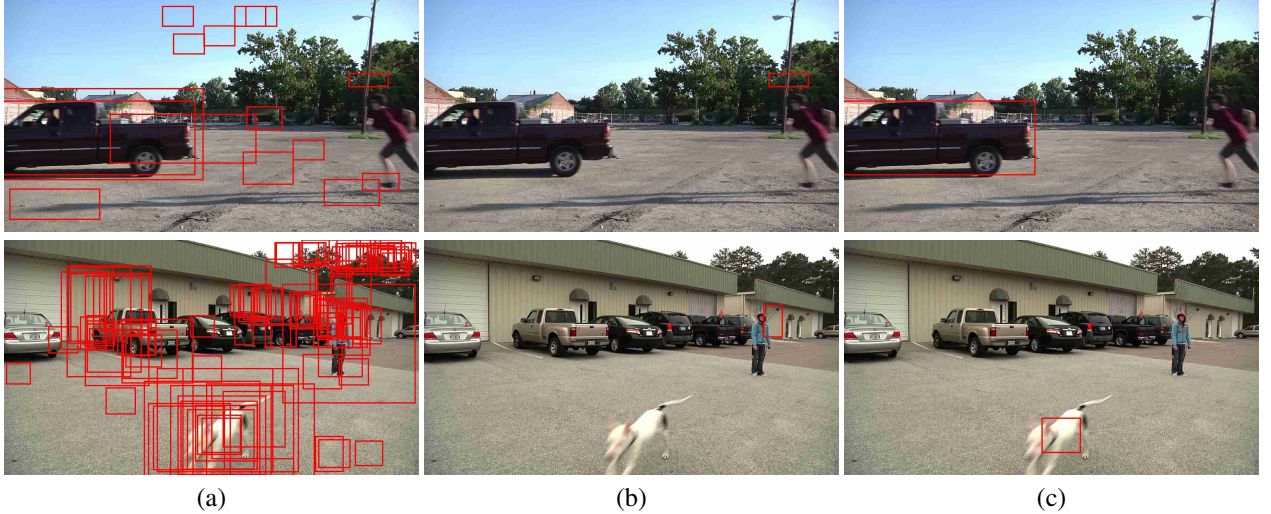


Figure 5: Improved performance of simultaneous object detection, tracking, and event recognition. (a) Output of the Felzenszwalb et al. detector. (b) Tracks produced by detection-based-tracking. (c) Tracks produced by simultaneous object-detection, tracking, and event recognition.

models were trained on isolated frames using the standard Felzenszwalb training software. The event models were trained using tracks produced by the detection-based tracking method described in Section 2. It is difficult to track the person running with detection-based tracking alone due to articulated appearance change and motion blur. Imposing the prior of detecting *running* biases the tracker to find the desired track.

Figure 5 demonstrates improved performance of simultaneous object detection, tracking, and event recognition (c) over object detection (a) and tracking (b) in isolation. As before, these results were obtained with object and event models that were trained independently.

8 Conclusion

Detection-base tracking using dynamic programming has a long history (Castanon, 1990; Wolf et al., 1989), as do motion-profile-based approaches to event recognition using HMMs (Siskind and Morris, 1996; Starner et al., 1998; Wang et al., 2009; Xu et al., 2002, 2005). Moreover, there have been attempts to integrate object detection and tracking (Li and Nevatia, 2008; Pirsiavash et al., 2011), tracking and event recognition (Li and Chellappa, 2002), and object detection and event recognition (Gupta and Davis, 2007; Moore et al., 1999; Peursum et al., 2005). However, we are unaware of prior work that integrates all three and does so in a fashion that efficiently finds a global optimum to a simple unified cost function.

We have demonstrated a general framework for simultaneous object detection, tracking, and event recognition. Many object detectors can naturally be transformed into trackers

by introducing time into their cost functions, thus tracking every possible detection in each frame. Furthermore, the distance transform can be used to reduce the complexity of doing so from quadratic to linear. The common internal structure and algorithmic organization of object detection, detection-based tracking, and event recognition further allows an HMM-based approach to event recognition to be incorporated into the general dynamic-programming approach. This facilitates multidirectional information flow where not only can object detection influence tracking and, in turn, event recognition, event recognition can influence tracking and, in turn object detection.

Acknowledgments

This work was supported, in part, by NSF grant CCF-0438806, by the Naval Research Laboratory under Contract Number N00173-10-1-G023, by the Army Research Laboratory accomplished under Cooperative Agreement Number W911NF-10-2-0060, and by computational resources provided by Information Technology at Purdue through its Rosen Center for Advanced Computing. Any views, opinions, findings, conclusions, or recommendations contained or expressed in this document or material are those of the author(s) and do not necessarily reflect or represent the views or official policies, either expressed or implied, of NSF, the Naval Research Laboratory, the Office of Naval Research, the Army Research Laboratory, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes, notwithstanding any copyright notation herein.

References

- L. E. Baum. An inequality and associated maximization technique in statistical estimation of probabilistic functions of a Markov process. *Inequalities*, 3:1–8, 1972.
- L. E. Baum and T. Petrie. Statistical inference for probabilistic functions of finite state Markov chains. *Ann. Math. Stat.*, 37: 1554–63, 1966.
- L. E. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *The Annals of Mathematical Statistics*, 41(1):164–71, 1970.
- H. Bui. personal communication, 2011.
- D.A. Castanon. Efficient algorithms for finding the K best paths through a trellis. *IEEE Transactions on Aerospace and Electronic Systems*, 26(2):405–10, March 1990.
- D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):564–75, 2003.
- J. Corso, 2011. URL http://www.cse.buffalo.edu/~jcorso/bigshare/mindseye_human_annotation_may11_buffalo.tar.bz.
- M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes (VOC) challenge. *International Journal of Computer Vision*, 88(2): 303–38, 2010.
- P. F. Felzenszwalb and D. P. Huttenlocher. Distance transforms of sampled functions. Technical Report TR2004-1963, Cornell Computing and Information Science, 2004.
- P. F. Felzenszwalb, R. B. Girshick, and D. McAllester. Cascade object detection with deformable part models. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010a.
- P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9), September 2010b.
- Pedro F. Felzenszwalb, Daniel P. Huttenlocher, and Jon M. Kleinberg. Fast algorithms for large-state-space HMMs with applications to web usage analysis. In *Neural Information Processing Systems*, 2003.
- W. T. Freeman and M. Roth. Orientation histograms for hand gesture recognition. In *International Workshop on Automatic Face and Gesture Recognition*, pages 296–301, June 1995.
- A. Gupta and L. S. Davis. Objects in action: an approach for combining action understanding and object perception. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2007.
- Baoxin Li and Rama Chellappa. A generic approach to simultaneous tracking and verification in video. *IEEE Transactions on Image Processing*, 11(5):530–44, 2002.
- Yuan Li and Ramakant Nevatia. Key object driven multi-category object recognition, localization, and tracking using spatio-temporal context. In *Proceedings of the European Conference on Computer Vision*, volume IV, pages 409–22, 2008.
- D. J. Moore, I. A. Essa, and M. H. Heyes. Exploiting human actions and object context for recognition tasks. In *Proceedings of the 7th International Conference on Computer Vision*, 1999.
- R. Navatia. personal communication, 2011.
- N. Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man and Cybernetics*, 9(1):62–6, January 1979. ISSN 0018-9472.
- P. Peursum, G. West, and S. Venkatesh. Combining image regions and human activity for indirect object recognition in indoor wide-angle views. In *Proceedings of the 10th International Conference on Computer Vision*, 2005.
- H. Pirsiavash, D. Ramanan, and C. C. Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of objects. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1201–8, 2011.
- K. Saenko, 2011. URL https://s3.amazonaws.com/Annotations/vaticlabels_C-D1_0819.tar.gz.
- Pablo Sala, Diego Macrini, and Sven J. Dickinson. Spatiotemporal contour grouping using abstract part models. In *Proceedings of the 10th Asian Conference on Computer Vision*, volume 4, pages 539–52, 2010.
- J. Shi and C. Tomasi. Good features to track. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 593–600, 1994.
- J. M. Siskind and Q. Morris. A maximum-likelihood approach to visual event classification. In *Proceedings of the Fourth European Conference on Computer Vision*, pages 347–60, April 1996.
- Thad Starner, Joshua Weaver, and Alex Pentland. Real-time American sign language recognition using desk and wearable computer based video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12):1371–5, 1998.
- C. Tomasi and T. Kanade. Detection and tracking of point features. Technical Report CMU-CS-91-132, Carnegie Mellon University, 1991.
- A. J. Viterbi. Convolutional codes and their performance in communication systems. *IEEE Transactions on Communication*, 19:751–72, October 1971.
- Zhaowen Wang, Ercan E. Kuruoglu, Xiaokang Yang, Yi Xu, and Songyu Yu. Event recognition with time varying hidden Markov model. In *Proceedings of the International Conference on Acoustic and Speech Signal Processing*, pages 1761–4, 2009.
- J. K. Wolf, A.M. Viterbi, and G. S. Dixon. Finding the best set of K paths through a trellis with application to multitarget tracking. *IEEE Transactions on Aerospace and Electronic Systems*, 25(2):287–96, March 1989.
- Gu Xu, Yu-Fei Ma, HongJiang Zhang, and Shiqiang Yang. Motion based event recognition using HMM. In *Proceedings of the International Conference on Pattern Recognition*, volume 2, 2002.
- Gu Xu, Yu-Fei Ma, HongJiang Zhang, and Shi-Qiang Yang. An HMM-based framework for video semantic analysis. *IEEE Trans. Circuits Syst. Video Techn.*, 15(11):1422–33, 2005.
- A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Computing Surveys*, 38(4), December 2006.